LA-10801-MS

UC-11 and UC-32
Issued: August 1986

BSTLC: A Computer Code to Locate Rockbursts

Edward M. Van Eeckhout*

*Long-Term Visiting Staff Member at Los Alamos. Mining Department, Montana College of Mineral Science and Technology, Butte, MT 59701.

Los Alamos
Los Alamos National Laboratory
Los Alamos, New Mexico 87545

CONTENTS

LIST OF TABLES

BSTLC: A COMPUTER CODE TO LOCATE ROCKBURSTS

by

Edward M. Van Eeckhout

ABSTRACT

The computer code BSTLC was written to locate rockbursts using five or more "hits" experienced by a geophone array. The code is based on solving a set of equations nonlinearly as each new hit is added to the data mix. Field trials of the code at the Sunshine Mine, Kellogg, Idaho, have shown that rockbursts can be located within 100 ft. The error encountered appears to be due to geologic and mining variables.

I. INTRODUCTION

In December 1984, the Sunshine Mining Company of Kellogg, Idaho, requested assistance in re-establishing its rockburst monitoring system. To accomplish that purpose, one graduate student from Montana Tech (John Jordon) was stationed on site to work on field installations. Additionally, this author wrote a computer code that would calculate burst locations given field data input from a geophone array (that is, time of hit and location of geophone). This report summarizes the programming effort and compares the location results of the new code (BSTLC) with those of other codes that have been used in the mining district. Appendix A gives user directions for BSTLC; Appendix B gives a listing of the code.

II. SUMMARY

The FORTRAN-based BSTLC can calculate rockburst locations from five or more hits to a geophone array. Test results have indicated that the code calculates burst locations as well as or better than any other code available, although the average location error experienced in the Sunshine Mine is still about 100 ft.

Some questions remain unanswered and require further investigation:

How does the geophone location affect solutions (i.e., planar versus spherical array)?

What are the best ways to throw away erroneous data points?

Why are the calculated burst locations still 100 ft off on the average?

III. BACKGROUND

Monitoring rockburst locations, both for safety and for predicting regions of instability, has become common in the Coeur d'Alene District of Idaho (see, for example, Blake, 1983). Most mines use these monitoring systems to locate bursts "after the fact," not as a tool to predict burst-prone areas. A great deal of effort has been expended in predicting damaging bursts from microseismic percursors (see Carlisle, 1983, and Langstaff, 1979), and it is clear that more work needs to be done on this aspect of the problem. Such work, which requires more resources than individual mines have at their disposal, has not received a great deal of attention recently.

This report deals specifically with the development of a new computer code to locate bursts in a mining situation--implementation of the code is described by Van Eeckhout et al., 1986.

A. Previous Codes

Codes commonly used for rockburst locations are based on those summarized first by Blake et al. (1974) and include the codes DIRSOLC, DIRSQRO, XYZVSE, and GBLK. The first is a P-wave direct solution for a burst location given five hits and a known velocity; the second and third are least-squares solutions for more than five hits; and the fourth is a generated block method that compares hit sequence with theoretical hit sequences given a certain source block. A more recent description of the use of these codes in mining is given in Blake (1982).

The problems one encounters using these codes are not great, and they have generally given good results depending on the application. However, these codes subtract the equation involving the first phone from the rest, making a set of linear equations from the nonlinear ones. Thus the least-squares solution tends to reduce the solution by one degree of freedom. It seems more appropriate to solve the entire set of equations using a nonlinear approach, allowing the solution to "float" more freely, and that is the intent of the code developed here. A somewhat similar approach has been described by Redfern and Munson (1982), but it is not yet widely used in the industry.

B. BSTLC Development

BSTLC is a code that solves a set of nonlinear equations (with the following unknown: time to first phone; gt; Cartesian coordinates gx, gy, and gz; and velocity v) by converging to a solution from a "first guess." It approaches the solution in a manner similar to the Newton-Raphson method of finding solutions for a single nonlinear equation, only with five unknowns--not one. Thus the coding reduces to setting up convergence techniques for an exactly determined (five equations, five unknowns) or an overdetermined (six or more equations, five unknowns) set of equations, as well as finding suitable first guesses. After that, various "filtering" techniques may be used to drop phones that do not fit the pattern (however that may be defined).

The equation being solved is

$$(t_i + gt) = \mathrm{sqrt}\left\{(x_i - gx)^2 + (y_i - gy)^2 + (z_i - gz)^2\right\}/v, \tag{1}$$

where

t_i = arrival time at geophone i (with all phones being relative to the first hit),

gt = unknown arrival time from the burst to the first phone hit,

x_i, y_i, z_i = locations of i^{th} geophone,

gx, gy, gz = unknown location of burst, and

v = unknown velocity.

Thus, there are five unknowns: gt, gx, gy, gz, and v.

If Eq. (1) is redefined as

$$F = (t_i + gt) - \mathrm{sqrt}\left\{(x_i - gx)^2 + (y_i - gy)^2 + (z_i - gz)^2\right\}/v \tag{2}$$

and if each i^{th} equation is set equal to zero for each of five hits in the exactly determined case, an iterative technique will find a solution (if the first guess is close enough) by updating the guess as follows:

gt = gt - s(1),
gx = gx - s(2),
gy = gy - s(3),
gz = gz - s(4), and
v = v - s(5).

{s} is an array of solutions to the following set of equations:

$$\begin{bmatrix} F_{t1} & F_{x1} & F_{y1} & F_{z1} & F_{v1} \\ F_{t2} & F_{x2} & F_{y2} & F_{z2} & F_{v2} \\ F_{t3} & F_{x3} & F_{y3} & F_{z3} & F_{v3} \\ F_{t4} & F_{x4} & F_{y4} & F_{z4} & F_{v4} \\ F_{t5} & F_{x5} & F_{y5} & F_{z5} & F_{v5} \end{bmatrix} \begin{Bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \end{Bmatrix}, \tag{3}$$

where

F_{ij} = the i^{th} partial derivative of F for the j^{th} phone,
S_i = adjustment to the i^{th} value in the solution, and
F_j = the value of F [Eq. (2)] for the j^{th} phone.

Introductory details to this method are discussed in James et al. (1977, pp. 271-273) and Dahlquist and Bjorck (1974, pp. 248-253). The coding of this technique is in subroutine NEWT and SETUP (see Appendix B). The matrices are solved using a Gaussian elimination technique with back substitution (in subroutine Gauss, Appendix B). This method appears to work well, unless the matrices are too ill conditioned (i.e., the phones lie near a single plane) or the solution is unreasonable. The program will only iterate 20 times, and then it will print ***SOLUTION DIVERGING***.

If more than five hits are experienced, the set of equations becomes over-determined and can be solved by minimizing the residuals in a least-squares manner, by using a linearized approximation. In this case, the adjustment to each iteration can be written as follows (Dahlquist and Bjorck, 1974, p. 443):

$$[F']^T[F'] \{s\} = [F']^T \{F\} , \tag{4}$$

where

$$[F']^T = \begin{bmatrix} F_{t1} & F_{t2} & \cdots & F_{tm} \\ F_{x1} & F_{x2} & \cdots & F_{xm} \\ F_{y1} & F_{y2} & \cdots & F_{ym} \\ F_{z1} & F_{z2} & \cdots & F_{zm} \\ F_{v1} & F_{v2} & \cdots & F_{vm} \end{bmatrix} ,$$

F_{ij}, S_i, F_j = as before, and m = number of phones hit.

Again, the coding of this method is in subroutine NEWT and SETUP, and all contributions over icount equations (icount is simply a counter for the number of equations) must be added and then solved.

The first problem with any iterative solution is the first guess. If that guess is too far away from the solution, the approximations to the solution may diverge. Two choices for the first guess have been incorporated into BSTLC: one choice is based on averaging all phone locations hit within 10 ms (subroutine GUESS1), and one is based on a direct solution of five phone hits, which was suggested by Blake et al. (1974, pp. 37-39) (subroutine GUESS2). The first alternative guesses gt = 0.05 s and v = 15000 ft/s and then averages x_i, y_i, and z_i for all phones hit in the first 10 ms. The second alternative fixes a velocity of 15000 ft/s, allows gt to "float," and solves directly for first guesses gt, gx, gy, and gz.

The next concern is to find some method to filter "bad" phones (that is, those phones that may not fall into the solution sequence very well, but were tripped inadvertently, or those phones that received a signal through a quite different material, such as fill). The assumption made in the development of BSTLC was that the closer the phone is to the event, the better the chances of a good "clear" hit. (That is, we hope not too much fill or other material will interfere.) This assumption will not always be true, but it seems to be a good starting point. Thus, the set of equations is solved for each new phone added to the matrix.

BSTLC has two filtering techniques incorporated into it; both are contained in the main part of this code. The first technique simply drops a phone from inclusion when the addition of that phone causes a divergence from a previous solution. The second technique starts by solving all sets of equations (when there are more than six phones) and then dropping one phone at a time. The standard deviation of the differences from the fit value is calculated for each set [$s = \sqrt{\text{variance}/(n - 5)}$], and the set of phones with the least deviation (the best fit) is compared with the deviation for all phones. If that deviation (called difdev) is greater than the t-statistic value times difmin, then the phone causing the greatest change (itemp) is discarded. For example, if the deviation of eight phones is 15 ft, the minimum deviation for seven phones is 1.2 ft, and the t-statistic for 2 degrees of freedom (7 minus 5) is 9.925 at 99% confidence (Blank, 1980, p. 630). Then the jump from 1.2 x 9.925 = 11.9 to 15 is too great and the bad phone is discarded.

IV. CODE VALIDATION

The first field test of BSTLC was performed at the Sunshine Mine, Kellogg, Idaho. A series of five test blasts were monitored using a network of 48 geophones. Locations of all phones are indicated in Table I, which is really a print-out of the GEOPH.DAT file. The identification number of the phone is first, and then x, y, z coordinates are listed for each phone. Because of the tabular nature of the orebody, the phones were located in a somewhat planar distribution pattern.

Table II lists the sequence of hits and times for Events (test blasts) 1 through 5. The actual locations, along with the BuMines least-squares solution for each event, are given in Table III. These solutions were provided from the site; no reasons can be found to explain why the velocities calculated are so inaccurate. In general, the solutions were from 50 to 500 ft off (with an average of 165 ft). Error was probably due to a number of factors: (1) the variable geology (a constant velocity is assumed), (2) some questions involving the triggering level of the geophones, and (3) errors in surveying blasts and geophones. But will BSTLC do any better?

Table IV lists the event locations plus the BSTLC location using filtering. The average error in this tabulation is 100 ft, which is still not particularly satisfying, but it is better.

Table V lists output for Event 1. Note that phone 9 was thrown out because it caused divergence after one good solution had been obtained. Nothing particularly remarkable was noted after that.

Table VI lists output for Event 2. Note that a solution was finally obtained after phone 6 was hit. The residuals (difdev) remained quite stable. However, Table VII shows what might happen with a bad guess. The initial direct solution is too far off and causes divergence. In general, the closest geophone method of guessing (subroutine GUESS1) was found to be better than the direct solution method (GUESS2).

Table VIII gives detail on filtering Event 3, the one event that really initiated the filtering portion of the code. Note how a deviation near zero was found for six phones and how adding a seventh generally put the deviation outside the tolerable level. Only one subsequent phone was allowed. The solution was fairly accurate.

Table IX gives the solutions for Event 4. A nice orderly progression is noted here.

Table X gives the solutions for Event 5. Note the jump in difdev. A smaller confidence interval might throw out a subsequent phone, which could be the basis for a slightly better solution. That possibility awaits further case examples.

TABLE I

GEOPHONE LOCATIONS

50,89,143,-1687	24,804,251,-2691
1,741,372,-1687	25,1058,774,-2690
2,445,652,-1885	26,1398,686,-2683
3,1033,279,-1886	27,-1011,378,-1152
4,561,189,-1885	28,-551,396,-1133
5,1048,137,-1892	29,-400,224,-1145
6,1105,185,-2085	30,-169,213,-1146
7,250,333,-2078	31,-115,97,-1145
8,593,697,-2086	32,88,21,-1144
9,936,252,-2085	33,-1121,434,-1002
10,968,393,-2085	34,-894,387,-1004
11,1417,277,-2083	35,-804,439,-1000
12,317,500,-2282	36,-762,201,-999
13,635,650,-2280	37,-572,222,-1001
14,898,520,-2283	38,-295,284,-999
15,1023,284,-2284	39,-285,103,-999
16,1161,358,-2284	40,-110,172,-997
17,1418,532,-2283	41,204,302,-993
19,310,743,-2480	42,0,31,0
20,836,710,-2490	43,-1205,282,-702
21,1109,405,-2489	44,-1018,355,-711
22,1186,537,-2486	45,-891,204,-703
23,437 816,-2682	46,-609,80,-699
	47,-616,209,-706

TABLE II

HIT TIMES FOR EVENTS 1 THROUGH 5

Event 1	Event 2	Event 3	Event 4	Event 5
1,0.	1,0.	10,0.	17,0.	39,0.
3,.0012	3,.00335	3,.00205	11,.00305	38,.0015
5,.00205	5,.0047	9,.0037	16,.0075	40,.0076
4,.009	10,.0113	1,.0059	22,.01375	29,.0084
10,.00915	9,.01275	5,.0061	6,.01475	30,.0104
9,.0107	4,.0128	14,.01045	15,.016	31,.01375
6,.0132	6,.01555	6,.01065	21,.01625	47,.01425
15,.02415	15,.02565	8,.0133	3,.0216	46,.0238
11,.02645	2,.0257	15,.01615	9,.0217	41,.04255
2,.02745	8,.0282	13,.0188	14,.0227	50,.0881
14,.0297	14,.0312	2,.0197	5,.0259	
8,.0317	11,.0329	11,.02205	20,.03195	
7,.0369	50,.0385	4,.02305	24,.03465	
21,.0391	21,.0409	20,.0257	25,.03705	
22,.04225	24,.043	22,.0285	8,.04035	
20,.04485	22,.0443	21,.0292		

TABLE III

EVENT LOCATIONS VERSUS BUMINES LEAST-SQUARES (LSSQ) SOLUTION

Event No.		x	y	z	v	Error	Number of Phones Used
1	Actual	852	306	-1811	-	-	-
	LSSQ	888	216	-1704	12840	144	16
2	Actual	840	315	-1811	-	-	-
	LSSQ	824	208	-1801	17590	109	16
3	Actual	852	418	-1950	-	-	-
	LSSQ	905	449	-1948	4040	61	16
4	Actual	1279	417	-2218	-	-	-
	LSSQ	843	288	-2113	28800	462	16
5	Actual	-375	138	-919	-	-	-
	LSSQ	-314	167	-911	5440	51	10

TABLE IV

EVENT LOCATIONS VERSUS BSTLC SOLUTION

Event No.		x	y	z	v	Error	Number of Phones Used
1	Actual	852	306	-1811	-	-	-
	BSTLC	904	225	-1718	15140	134	15
2	Actual	840	315	-1811	-	-	-
	BSTLC	856	245	-1744	16220	98	16
3	Actual	852	418	-1950	-	-	-
	BSTLC	885	436	-1926	17300	45	7
4	Actual	1279	417	-2218	-	-	-
	BSTLC	1396	406	-2209	27560	118	16
5	Actual	-375	138	-919	-	-	-
	BSTLC	-319	189	-876	8450	87	10

TABLE V

BSTLC OUTPUT FOR EVENT 1, USING NO FILTERING AND GUESS1

DATA FILE USED. f1d101 dat

NO FILTER USED

INITIAL GUESS: using ave of nearest phones (t & v fixed)

t : 0 050 x = 871 y = 277 z = -1888 v = 15000

SOLUTIONS:

T	X	Y	Z	V	DIFDEV	PHONES USED
0.01265	1011	252	-1510	27171	0.0	1, 3, 5, 4,10,
	*** DIVERGING SOLUTION ***					1, 3, 5, 4,10, 9,
PHONE 9 IS OUT						
0.01226	869	252	-1775	15888	27 4	1, 3, 5, 4,10, 6,
0 01514	856	249	-1800	13266	21.2	1, 3, 5, 4,10, 6,15,
0.01303	881	244	-1745	15777	27 4	1, 3, 5, 4,10, 6,15,11,
0.01341	884	237	-1747	15420	24.6	1, 2, 5, 4,10, 6,15,11, 2,
0 01479	890	216	-1719	15192	29.9	1, 3, 5, 4,10, 6,15,11, 2,14,
0.01533	883	207	-1715	14996	29 1	1, 3, 5, 4,10, 6,15,11, 2,14, 8,
0.01552	899	221	-1719	14644	27.9	1, 3, 5, 4,10, 6,15,11, 2,14, 8, 7,
0.01519	901	221	-1720	14818	27.0	1, 3, 5, 4,10, 6,15,11, 2,14, 8, 7,21,
0 01489	904	222	-1716	15042	26.8	1, 3, 5, 4,10, 6,15,11, 2,14, 8, 7,21,22,
0.01461	904	225	-1718	15137	26.0	1, 3, 5, 4,10, 6,15,11, 2,14, 8, 7,21,22,20,

Phones not used in last solution given: 9,

TABLE VI

BSTLC OUTPUT FOR EVENT 2, USING NO FILTERING AND GUESS1

DATA FILE USED: f1d102.dat

NO FILTER USED

INITIAL GUESS: using ave of nearest phones (t & v fixed)

t : 0.050 x = 942 y = 266 z = -1823 v = 15000

SOLUTIONS:

T	X	Y	Z	V	DIFDEV	PHONES USED
	*** DIVERGING SOLUTION ***					1, 3, 5,10, 9,
	*** DIVERGING SOLUTION ***					1, 3, 5,10, 9, 4,
0.01092	928	302	-1651	18884	28 3	1, 3, 5,10, 9, 4, 6,
0.01166	902	283	-1704	16240	24.3	1, 3, 5,10, 9, 4, 6,15,
0.01153	905	285	-1696	16517	21 5	1, 3, 5,10, 9, 4, 6,15, 2,
0 01165	904	282	-1697	16402	19 4	1, 3, 5,10, 9, 4, 6,15, 2, 8,
0 01307	907	266	-1671	16073	29 7	1, 3, 5,10, 9, 4, 6,15, 2, 8,14,
0.01241	872	280	-1739	14829	28 7	1, 3, 5,10, 9, 4, 6,15, 2, 8,14,11,
0 01215	867	261	-1722	15262	29.7	1, 3, 5,10, 9, 4, 6,15, 2, 8,14,11,50,
0.01204	867	261	-1723	15317	28.2	1, 3, 5,10, 9, 4, 6,15, 2, 8,14,11,50,21,
0.00975	859	248	-1742	16380	44 3	1, 3, 5,10 9, 4, 6,15, 2, 8,14,11,50,21,24.
0 00999	856	245	-1744	16216	42.2	1, 3, 5,10, 9, 4, 6,15, 2, 8,14,11,50,21,24,22,

TABLE VII

BSTLC OUTPUT FOR EVENT 2, USING NO FILTERING AND GUESS2

```
DATA FILE USED:     fld102.dat
NO FILTER USED
INITIAL GUESS:  using an initial direct solution (v fixed)
     t =  0.152  x =  1792  y =   739  z =   303  v = 15000
SOLUTIONS:
     T          X          Y          Z          V        DIFDEV     PHONES USED
             *** DIVERGING SOLUTION ***                                1, 3, 5,10, 9,
             *** DIVERGING SOLUTION ***                                1, 3, 5,10, 9, 4,
             *** DIVERGING SOLUTION ***                                1, 3, 5,10, 9, 4, 6,
             *** DIVERGING SOLUTION ***                                1, 3, 5,10, 9, 4, 6,15,
             *** DIVERGING SOLUTION ***                                1, 3, 5,10, 9, 4, 6,15, 2,
             *** DIVERGING SOLUTION ***                                1, 3, 5,10, 9, 4, 6,15, 2, 8,
             *** DIVERGING SOLUTION ***                                1, 3, 5,10, 9, 4, 6,15, 2, 8,14,
             *** DIVERGING SOLUTION ***                                1, 3, 5,10, 9, 4, 6,15, 2, 8,14,11,
             *** DIVERGING SOLUTION ***                                1, 3, 5,10, 9, 4, 6,15, 2, 8,14,11,50,
             *** DIVERGING SOLUTION ***                                1, 3, 5,10, 9, 4, 6,15, 2, 8,14,11,50,21,
             *** DIVERGING SOLUTION ***                                1, 3, 5,10, 9, 4, 6,15, 2, 8,14,11,50,21,24,
             *** DIVERGING SOLUTION ***                                1, 3, 5,10, 9, 4, 6,15, 2, 8,14,11,50,21,24,22,
```

TABLE VIII

BSTLC OUTPUT FOR EVENT 3, USING FILTERING AND GUESS1

```
DATA FILE USED:     fld103.dat

FILTER USED

INITIAL GUESS:  using ave of nearest phones (t & v fixed)

     t =  0.050  x =   946  y =   290  z = -1928  v = 15000

SOLUTIONS:

    T            X          Y          Z          V        DIFDEV     PHONES USED

             *** DIVERGING SOLUTION ***                               10, 3, 9, 1, 5,

  0.00722       905        408      -1921      21868       32.9      10, 3, 9, 1, 5,14,

  0.00686       881        396      -1915      21740       42.6       3, 9, 1, 5,14, 6,
  0.01049       921        434      -1905      18085       35.6      10, 9, 1, 5,14, 6,
  0.01239       823        346      -1960      16010       30.1      10, 3, 1, 5,14, 6,
  0.01286       825        427      -1879      18247       34.2      10, 3, 9, 5,14, 6,
  0.01055       885        437      -1925      17440        0.0      10, 3, 9, 1,14, 6,
  0 00918       887        405      -1919      18575       40.6      10, 3, 9, 1, 5, 6,
  0.00888       890        415      -1920      19338       30.3      10, 3, 9, 1, 5,14, 6,

 DIFDEV =     30.3  ALLOWABLE INTERVAL =        1.6

 PHONE  5 IS OUT

  0.01055       885        437      -1925      17440        0.0      10, 3, 9, 1,14, 6,
  0.01049       884        439      -1925      17590        3.9       3, 9, 1,14, 6, 8,
  0.01044       881        437      -1927      17587        3.4      10, 9, 1,14, 6, 8,
  0.01057       890        445      -1922      17608        3.5      10, 3, 1,14, 6, 8,
  0.01077       881        438      -1923      17438        1.7      10, 3, 9,14, 6, 8,
  0.01052       886        441      -1926      17616        2.1      10, 3, 9, 1, 6, 8,
  0.01052       884        439      -1925      17572        3.9      10, 3, 9, 1,14, 8,
  0.01053       884        439      -1925      17556        2.7      10, 3, 9, 1,14, 6, 8,

 DIFDEV =      2.7  ALLOWABLE INTERVAL =        1.6

 PHONE  8 IS OUT

  0.01055       885        437      -1925      17440        0.0      10, 3, 9, 1,14, 6,
  0.01443       892        456      -1923      14411       30.0       3, 9, 1,14, 6,15,
  0.01349       906        458      -1910      15160       32.4      10, 9, 1,14, 6,15,
  0.01611       970        541      -1877      15475       28.0      10, 3, 1,14, 6,15,
  0.01458       855        451      -1898      15409       32.0      10, 3, 9,14, 6,15,
  0.01227       881        411      -1918      14941       23.9      10, 3, 9, 1, 6,15,
  0.01459       873        455      -1925      14106       20.5      10, 3, 9, 1,14,15,
  0.01283       888        448      -1919      15379       23.9      10, 3, 9, 1,14, 6,15,

 DIFDEV =     23.9  ALLOWABLE INTERVAL =        1.6

 PHONE 15 IS OUT

  0.01055       885        437      -1925      17440        0.0      10, 3, 9, 1,14, 6,
  0.01190       892        439      -1925      16066       14.4       3, 9, 1,14, 6,13,
  0.01158       904        442      -1916      16558       12.7      10, 9, 1,14, 6,13,
  0.01119       883        425      -1927      16478       15.6      10, 3, 1,14, 6,13,
  0.00986       907        433      -1937      17155       10.4      10, 3, 9,14, 6,13,
  0.01092       884        423      -1923      16587        8.2      10, 3, 9, 1, 6,13,
  0.01201       885        439      -1925      15877       13.6      10, 3, 9, 1,14,13,
  0.01119       890        434      -1924      16575       11.1      10, 3, 9, 1,14, 6,13,


 DIFDEV =     11.1  ALLOWABLE INTERVAL =        1.6

 PHONE 13 IS OUT

  0.01055       885        437      -1925      17440        0.0      10, 3, 9, 1,14, 6,
  0.01232       894        442      -1928      15749       14.8       3, 9, 1,14, 6, 2,
  0.01173       907        443      -1917      16394       14.9      10, 9, 1,14, 6, 2,
  0.01153       864        402      -1941      15996       16.6      10, 3, 1,14, 6, 2,
  0.00982       904        434      -1937      17348        7.5      10, 3, 9,14, 6, 2,
  0.01104       885        420      -1923      16342       10.6      10, 3, 9, 1, 6, 2,
  0.01204       887        439      -1927      15772       16.1      10, 3, 9, 1,14, 2,
  0.01123       893        434      -1926      16462       12.6      10, 3, 9, 1,14, 6, 2,

 DIFDEV =     12.6  ALLOWABLE INTERVAL =        1.6

 PHONE  2 IS OUT
```

TABLE VIII (cont)

0.01055	885	437	-1925	17440	0.0	10, 3, 9, 1,14, 6,
0.01019	887	437	-1924	17824	2.8	3, 9, 1,14, 6,11,
0.01046	890	439	-1923	17611	2.9	10, 9, 1,14, 6,11,
0.01052	892	444	-1922	17600	1.8	10, 3, 1,14, 6,11,
0.01023	891	438	-1926	17679	3.0	10, 3, 9,14, 6,11,
0.01041	888	441	-1925	17720	2.4	10, 3, 9, 1, 6,11,
0.01025	888	436	-1925	17775	1.9	10, 3, 9, 1,14,11,
0.01035	888	438	-1924	17702	2.3	10, 3, 9, 1,14, 6,11,

DIFDEV = 2.3 ALLOWABLE INTERVAL = 1.6

PHONE 11 IS OUT

0.01055	885	437	-1925	17440	0.0	10, 3, 9, 1,14, 6,
0.02134	959	521	-1912	12073	46.4	3, 9, 1,14, 6, 4,
0.02256	1025	540	-1864	13027	42.1	10, 9, 1,14, 6, 4,
0.02477	1070	639	-1832	14353	28.5	10, 3, 1,14, 6, 4,
0.01054	994	461	-1961	15665	38.9	10, 3, 9,14, 6, 4,
0.02438	1040	640	-1884	14305	60.5	10, 3, 9, 1, 6, 4,
0.02104	942	530	-1902	12023	48.5	10, 3, 9, 1,14, 4,
0.01933	996	546	-1887	13839	42.7	10, 3, 9, 1,14, 6, 4,

DIFDEV = 42.7 ALLOWABLE INTERVAL = 1.6

PHONE 4 IS OUT

0.01055	885	437	-1925	17440	0.0	10, 3, 9, 1,14, 6,
0.01076	885	437	-1926	17234	1.4	3, 9, 1,14, 6,20,
0.01069	886	437	-1925	17294	1.6	10, 9, 1,14, 6,20,
0.01063	882	432	-1927	17304	1.3	10, 3, 1,14, 6,20,
0.01061	886	436	-1926	17312	1.7	10, 3, 9,14, 6,20,
0.01060	885	435	-1925	17335	0.9	10, 3, 9, 1, 6,20,
0.01074	884	437	-1926	17240	1.2	10, 3, 9, 1,14,20,
0.01067	885	436	-1926	17296	1.2	10, 3, 9, 1,14, 6,20,

DIFDEV = 1.2 ALLOWABLE INTERVAL = 1.6

0.01186	874	437	-1931	16265	14.3	3, 9, 1,14, 6,20,22,
0.01120	859	426	-1937	16616	14.0	10, 9, 1,14, 6,20,22,
0.01131	848	402	-1945	16615	12.5	10, 3, 1,14, 6,20,22,
0.01304	846	435	-1909	16444	11.0	10, 3, 9,14, 6,20,22,
0.01131	871	430	-1930	16605	14.9	10, 3, 9, 1, 6,20,22,
0.01183	870	440	-1930	16272	12.6	10, 3, 9, 1,14,20,22,
0.01165	877	431	-1928	16219	9.7	10, 3, 9, 1,14, 6,22,
0.01143	870	431	-1930	16537	12.2	10, 3, 9, 1,14, 6,20,22,

DIFDEV = , 12.2 ALLOWABLE INTERVAL = 12.0

PHONE 22 IS OUT

0.01067	885	436	-1926	17296	1.2	10, 3, 9, 1,14, 6,20,
0.01482	881	464	-1929	14503	33.1	3, 9, 1,14, 6,20,21,
0.01308	854	441	-1935	15250	36.1	10, 9, 1,14, 6,20,21,
0.01451	904	488	-1908	14979	35.6	10, 3, 1,14, 6,20,21,
0.01910	781	448	-1867	14931	26.0	10, 3, 9,14, 6,20,21,
0.01331	872	450	-1926	15243	36.5	10, 3, 9, 1, 6,20,21,
0.01451	866	469	-1927	14619	29.4	10, 3, 9, 1,14,20,21,
0.01364	873	431	-1928	14532	22.6	10, 3, 9, 1,14, 6,21,
0.01339	872	451	-1926	15206	29.8	10, 3, 9, 1,14, 6,20,21,

DIFDEV = 29.8 ALLOWABLE INTERVAL = 12.0

PHONE 21 IS OUT

0.01067	885	436	-1926	17296	1.2	10, 3, 9, 1,14, 6,20,

Phones not used in last solution given: 5, 8,15,13, 2,11, 4,22,21,

TABLE IX

BSTLC OUTPUT FOR EVENT 4, USING NO FILTERING AND GUESS1

DATA FILE USED: f1d104.dat

NO FILTER USED

INITIAL GUESS: using ave of nearest phones (t & v fixed)

t = 0.050 x = 1333 y = 392 z = -2218 v = 15000

SOLUTIONS:

T	X	Y	Z	V	DIFDEV	PHONES USED
0.00789	1399	441	-2181	17883	0.0	17,11,16,22, 8,
0.00777	1399	442	-2180	17782	3.1	17,11,16,22, 6,15,
0.00790	1396	446	-2175	17596	2.4	17,11,16,22, 6,15,21,
0.00797	1402	424	-2198	17186	5.6	17,11,16,22, 6,15,21, 3,
0.00798	1402	424	-2199	17178	4.5	17,11,16,22, 6,15,21, 3, 9,
0.00803	1404	420	-2200	17129	5.0	17,11,16,22, 6,15,21, 3, 9,14,
0.00871	1387	426	-2192	16122	10.1	17,11,16,22, 6,15,21, 3, 9,14, 5,
0.00793	1400	425	-2205	17072	12.3	17,11,16,22, 6,15,21, 3, 9,14, 5,20,
0.00737	1435	399	-2233	18410	17.6	17,11,16,22, 6,15,21, 3, 9,14, 5 20,24,
0.00766	1400	390	-2223	17042	32.5	17,11,16,22, 6,15,21, 3, 8,14, 5,20,24,25,
0.00697	1386	406	-2209	17563	34.5	17,11,16,22, 6,15,21, 3, 9,14, 5,20,24,25, 8,

TABLE X

BSTLC OUTPUT FOR EVENT 5, USING NO FILTERING AND GUESS1

DATA FILE USED: f1d105.dat

NO FILTER USED

INITIAL GUESS: using ave of nearest phones (t & v fixed)

t = 0.050 x = -271 y = 199 z = -1036 v = 15000

SOLUTIONS:

T	X	Y	Z	V	DIFDEV	PHONES USED
0.00677	-316	178	-938	14954	0.0	39,38,40,29,30,
0.00692	-315	180	-941	14543	4.8	39,38,40,29,30,31,
0.00826	-333	175	-904	15380	6.6	39,38,40,29,30,31,47,
0.00742	-317	194	-932	14484	51.3	39,38,40,29,30,31,47,46,
0.01344	-349	176	-931	10788	47.4	39,38,40,29,30,31,47,46,41,
0.02216	-319	185	-876	8447	44.1	39,38,40,29,30,31,47,46,41,50,

APPENDIX A: USER DIRECTIONS

The use of BSTLC is quite simple. First create a data file with at least five hits in a format similar to that shown in Table II. The program will interact with the terminal. For example, here is the response to create the output of Table V:

```
$ RUN BSTLC <cr>
Enter the input data file name:
FLD101.DAT <cr>
Enter the output data file name:
FLD101.DAT <cr>
Enter 1 for nearest geophone method of guess:
Enter 2 for direct solution guess:
1 <cr>
Enter 0 for no filtering,
Enter 1 for filtering:
0 <cr>
FORTRAN STOP
$
```

The prompts are in lower case printing; responses are in capital letters with a carriage return indicated (<cr>). The output is contained in FLD101.OUT.

APPENDIX B: CODE LISTING

```
C
C *** FILE BSTLC.CMN:
C
      implicit double precision (a-h),(o-z)
      parameter imax=50
      character*20 filein,fileout
      common a(5,6),s(5),n,np1,t(imax),x(imax),y(imax),
     .  z(imax),f(6),xloc(imax),yloc(imax),zloc(imax),
     .  torig(imax),id(imax),icount,gt,gx,gy,gz,v,
     .  gto,vo,idiv,idsave(imax),idout(imax),iout,
     .  filein,fileout,iguess,diff(imax),difdev,
     .  tstat(12),ihead,difmin,itemp,ifltr,iflag
C
      PROGRAM BSTLC
C
C *** VERSION 1.0
C
C *** This program was written by:
C
C                    Ed Van Eeckhout
C                    Mining Dept.
C                    Butte Mt 59701
C                    (406)496-4298
C
C *** LAST REVISED: June 1985
C
C *** VARIABLE IDENTIFICATION:
C
C       a . . . dummy arrays used to set up the 5X6 matrix
C       diff .  array to store calculated distance error of phone
C                 parameters vs burst location
C       difave  average of all phone differences in a particular solution
C                  (should be zero)
C       difdev  std dev of the differences
C       difmin  minimum standard deviation of differences for a set
C                 of solutions in subroutine FLTR
C       difsum  dummy variable for adding all differences
C       f . . . array to store partial derivatives in subroutine setup
C       gto . . original guess on time, to be used for each separate
C                  solution
C       gt . .  solution time to first geophone
C       gx . .  solution x-location of burst
C       gy . .  solution y-location of burst
C       gz . .  solution z-location of burst
C       icount  number of phones in a solution
C       id . .  array to store id number of all geophones
C       idiv .  equal to 1 if solution diverges (0 otherwise)
C       idout . array to store id number of geophones dropped from
C                  the solution
C       idsave  array to store id number of geophones saved in the
C                  cumulative solution
C       ifltr   dummy integer flag for filter
C       ihead . dummy integer flag for printing a header
C       n . . . dummy integer describing size of matrix to be solved
C       np1 . . n + 1, dummy integer for subroutine gauss
C       s . . . array of changes (from gauss) to be made to solutions
C       t . . . array describing 'hit' times to each geophone
C                 retained in the solution
C       torig . array describing 'hit' times to each geophone
C                 found from its id number
C       tstat . array of t-statistic values at the 99% confidence
C                 level
C       v . . . V, solution velocity
C       vo . .  original velocity guess
C       x . . . array describing x-location of each geophone
C                 retained in the cumulative solution
C       xloc .  array describing x-location of each geophone
C                 found from its id number
C       y . . . array describing y-location of each geophone
C                 retained in the cumulative solution
C       yloc .  array describing y-location of each geophone
C                 found from its id number
C       z . . . array describing z-location of each geophone
C               retained in the cumulative solution
C       zloc .  array describing z-location of each geophone
C                 found from its id number
C
C *** SUBROUTINE LIST:
C
C       GUESS1 . . . . Subroutine to guess an initial location for
```

```
C                         the burst using only phones with t < .01 sec.
C
C       GUESS2 . . . . Subroutine to guess an initial location for
C                         the burst using a direct solution.
C
C       HEADER . . . . Subroutine used to print headers for the output.
C
C       NEWT . . .,. . Subroutine to solve the set of equations
C                         iteratively (only 20 iterations allowed).
C
C       SETUP . . . .  Subroutine to set up the set of equations to
C                        be solved by NEWT.
C
C       GAUSS . . . .  Subroutine that solves a set of linear
C                        equations using Gaussian elimination with
C                        back-substitution.
C
C       SORT . . . . . Subroutine to find a set of 5 phones (out of
C                        6) that will converge, if needed.
C
C       DEVCAL . . . . Subroutine to calculate standard deviations
C                        (and solutions) of all mixes of phones by
C                        dropping one phone at a time.
C
C *** Input is effected both interactively and in a data file.
C       Interactive input required are the data file names and program
C       options (guess1 vs guess2 and filtering vs no filtering).  Prompt
C       commands will be displayed.  The data file input is quite
C       straightforward--simply enter each phone number plus the hit
C       time on one line in free format.
C
C *** Output is received in the data file named.
C
C *** The solution method employed is an iterative one based on
C       Taylor's series expansion of partial derivatives.  It is
C       similar to the Newton-Raphson method of finding roots of
C       polynomial equations, only it's applied to nonlinear
C       simultaneous equations in this case.  Two references to the
C       method as used for solving for 5 unknowns (t,x,y,z,and v)
C       given 5 'hits' are:
C
C         Dahlquist, G., Bjorck, A., 1974, Numerical Methods,
C           Prentice-Hall, Inc., pp. 248-253.
C
C         James,M.,Smith,G.,Wolford,J.,1977, Applied Numerical
C           Methods for Digital Computation, 2nd Ed., Harper and
C           Row, pp. 271-273.
C
C *** The solution method employed for solving overdetermined
C       nonlinear equations (number of 'hits' greater than 5)
C       is described in Dahlquist (1974, pp. 438-446) and is
C       called the Gauss-Newton method.
C
C *** The include statement is used here -- a listing of the
C       variables in common is given at the beginning of the
C       program listing
C
      include 'bstlc.cmn'
C
C *** Data statement setting matrix size (n,np1), t, v, and
C       tstat values (at the 99% confidence level)
C
      data n,np1,gt,gto,v,vo/5,6,.05,.05,15000.,15000./
      data tstat/63.657,9.925,5.841,4.604,4.032,3.707,3.499,
     *        3.355,3.250,3.169,3.106,3.055/
C
C *** Set input/output file names and optional items
C
      print *,' Enter the input data file name:'
      read (*,1001)filein
      print *,' Enter the output data file name:'
      read (*,1001)fileout
      print *,' Enter 1 for nearest geophone method of guess,
      print *,' Enter 2 for direct solution guess:'
      read (*,1003)iguess
      print *,' Enter 0 for no filtering,'
      print *,' Enter 1 for filtering:'
      read (*,1003)ifltr
      open(61,file='geoph.dat',status='old')
      open(71,file=filein,status='old')
```

```
      open(81,file=fileout,status='new')
C
C *** Read all geophone locations and save them for future use
C
    8 read(61,*,end=10)i,xloc(i),yloc(i),zloc(i)
      go to 8
C
C *** Read times for each burst--solve incrementally after the
C       first 5.
C
   10 read(71,*,end=300)i,torig(i)
      icount=icount+1
      id(icount)=i
      idsave(icount)=i
      t(icount)=torig(i)
      x(icount)=xloc(i)
      y(icount)=yloc(i)
      z(icount)=zloc(i)
C
C *** This if statement allows a new solution after each new
C       geophone time is read.
C
      if(icount.ge.5)go to 12
      go to 10
C
   12 continue
C
C *** Solve for the geophones used
C
      if(ifltr.eq.1.and.icount.gt.6)call devcal
C
      if(iguess.eq.1)call guess1
      if(iguess.eq.2)call guess2
      if(ihead.ne.1)call header
      call newt
      if(idiv.eq.1)difdev=5000.
C
C *** Drop a phone if adding it results in divergence (if iflag = 1,
C        at least one solution has been obtained).
C
      if(idiv.eq.0.and.iflag.eq.0)iflag=1
      if(icount.gt.5.and.idiv.eq.1.and.iflag.eq.1)then
        idiv=0
        iout=iout+1
        icount=icount-1
        idout(iout)=idsave(icount+1)
        write(81,*)
        write(81,1005)idout(iout)
        write(81,*)
        go to 10
      else
      endif
      write(81,*)
C
C *** This is the filter
C
      if(ifltr.eq.1.and.icount.gt.6)then
        write(81,1004)difdev,tstat(icount-6)*difmin
        write(81,*)
C
        if(difdev.lt.tstat(icount-6)*difmin)go to 10
C
        iout=iout+1
        icount=icount-1
        idout(iout)=idsave(itemp)
        write(81,1005)idsave(itemp)
        write(81,*)
C
        do 100 i = 1,icount
        ij=i
        if(itemp.le.i)ij=i+1
        idsave(i)=id(ij)
        id(i)=idsave(i)
        t(i)=torig(id(ij))
        x(i)=xloc(id(ij))
        y(i)=yloc(id(ij))
  100   z(i)=zloc(id(ij))
C
        if(iguess.eq.1)call guess1
        if(iguess.eq.2)call guess2
```

```
        call newt
      else
      endif
      go to 10
C
  300 continue
      if(iout.ge.1)write(81,1002)(idout(i),i=1,iout)
      stop
C
C *** FORMAT STATEMENTS
C
 1001 format(a20)
 1002 format(/'       Phones not used in last solution given:
     ,10(i2,',',))
 1003 format(i2)
 1004 format('  DIFDEV = ',F7.1,'  ALLOWABLE INTERVAL = ',F10.1)
 1005 format('  PHONE ',I2,' IS OUT')
      end
```

```
      SUBROUTINE GUESS1
C
C *** Subroutine to guess an initial location for the burst.  Use
C       only phones with t < .01 sec.
C
      include 'bstlc.cmn'
      xsum=0.
      ysum=0.
      zsum=0.
      kcount=0
C
C *** A do loop to add all phone locations that have a 'hit'
C       time of < .01 sec
C
      do 50 i=1,icount
        if (t(i).lt..01) then
          kcount=kcount+1
          xsum=xsum+x(i)
          ysum=ysum+y(i)
          zsum=zsum+z(i)
          else
          continue
        endif
   50 continue
C
C *** The values 1.,3.,-2. are just random so the first guess
C       does not sit exactly on only 1 geophone (causing a 0
C       divide in the solution).
C
      gx=xsum/real(kcount)+1.
      gy=ysum/real(kcount)+3.
      gz=zsum/real(kcount)-2.
      gt=gto
      v=vo
      return
      end
```

```
      SUBROUTINE GUESS2
C
C *** Subroutine to guess an initial solution using a direct solution
C       method described in USBM Bulletin 665, Appendix C.
C       All variables are defined in that publication.
C
      include 'bstlc.cmn'
      dimension locsq(5),d(5),alph(5),beta(5),gamm(5),l(5)
      real locsq,l
      v=vo
      do 10 I=1,5
      locsq(i)=x(i)**2+y(i)**2+z(i)**2
      d(i)=v*t(i)
      alph(i)=x(1)-x(i)
      beta(i)=y(1)-y(i)
      gamm(i)=z(1)-z(i)
   10 l(i)=d(i)**2+locsq(1)-locsq(i)
      do 20 k=3,5
      a(k-2,1)=2.*(alph(2)/d(2)-alph(k)/d(k))
      a(k-2,2)=2.*(beta(2)/d(2)-beta(k)/d(k))
      a(k-2,3)=2.*(gamm(2)/d(2)-gamm(k)/d(k))
   20 a(k-2,4)=l(2)/d(2)-l(k)/d(k)
      n=3
      np1=4
      call gauss
      gx=s(1)
      gy=s(2)
      gz=s(3)
      gt=sqrt((gx-x(1))**2+(gy-y(1))**2+(gz-z(1))**2)/v
      n=5
      np1=6
      return
      end
```

```
      SUBROUTINE HEADER
C
C *** Subroutine to print headers
C
      include 'bstlc.cmn'
      ihead=1
      write(81,1009)filein
      if(ifltr.eq.0)write(81,*)'NO FILTER USED'
      if(ifltr.eq.1)write(81,*)'FILTER USED'
      if(iguess.eq.1)write(81,1010)
      if(iguess.eq.2)write(81,1011)
      write(81,1012)gt,int(gx+.5),int(gy+.5),int(gz+.5),int(v+.5)
      return
C
C *** FORMAT STATEMENTS
C
 1009 format('1'/' DATA FILE USED:',5x,a20/)
 1010 format(/' INITIAL GUESS:  using ave of nearest phones '
     .          '(t & v fixed)'/)
 1011 format(/' INITIAL GUESS:  using an initial direct solution '
     .          '(v fixed)'/)
 1012 format(5x,'  t = ',f6.3,'  x = ',i5,'  y = ',i5,'  z = ',i5,
     .          '  v = ',i5//' SOLUTIONS:'//6x,'T',11x,'X',
     .         9x,'Y',9x,'Z',8x,'V',7x,'DIFDEV',5x,'PHONES USED'/)
      end
```

```
      SUBROUTINE NEWT
C
C *** Subroutine to solve the set of equations--if it doesn't converge
C       after 20 times, either the solution is unreasonable (outside
C       of the allowable range) or the matrix is too ill-conditioned
C
      include 'bstlc.cmn'
      do 100 k = 1,20
C
C *** Set up matrix to be solved by subroutine gauss.
C
      call setup
C
C *** Return changes to be applied to solutions
C
      call gauss
C
C *** Apply changes to solutions after each loop.  Solutions are
C       allowed in the following ranges:
C           gt . . . 0 to 2 sec
C           gx . . . -20000 to +20000 ft
C           gy . . . -20000 to +20000 ft
C           gz . . . -20000 to the surface (0)
C           v  . . . 3000 to 28000 ft/sec
C
      gt=gt-s(1)
      if(gt.gt.2..or.gt.lt.0.)gt=.05
      gx=gx-s(2)
      if(gx.gt.20000..or.gx.lt.-20000.)gx=0.
      gy=gy-s(3)
      if(gy.gt.20000..or.gy.lt.-20000.)gy=0.
      gz=gz-s(4)
      if(gz.gt.0..or.gz.lt.-20000.)gz=-5000.
      v=v-s(5)
      if(v.gt.28000..or.v.lt.3000.)v=5000.
C
C *** Test for convergence (changes in t .lt. .01 msec,
C       x,y,z .lt. 1 ft, and v .lt. 10 ft/sec)
C
  100 if (abs(s(1)).lt.1.e-5.and.abs(s(2)).lt.1..and.
             abs(s(3)).lt.1..and.abs(s(4)).lt.1..and.
         abs(s(5)).lt.10.)go to 200
      if (k.eq.21) then
        write(81,1002)(idsave(i),i=1,icount)
        idiv=1
        return
        else
      endif
  200 idiv=0
C
C *** A measure of the variance of the solution from all the
C       data points is the difdev , calculated here.  It is
C       a measure of cumulative error, in ft.
C
      difdev=0.
      difsum=0.
      if(icount.eq.5)go to 301
      do 300 i=1,icount
      diff(i)=(gt+t(i))*v-sqrt((gx-x(i))**2+(gy-y(i))**2
             +(gz-z(i))**2)
  300 difsum=difsum+diff(i)
      difave=difsum/float(icount)
      difsum=0.
      do 400 i=1,icount
  400 difsum=difsum+(diff(i)-difave)**2
      difdev=sqrt(difsum/(float(icount-5)))
  301 write(81,1001)gt,int(gx+.5),int(gy+.5),int(gz+.5),int(v+.5),
          difdev,(idsave(i),i=1,icount)
      return
C
C *** FORMAT STATEMENT
C
 1001 format(f10.5,4i10,f10.1,5x,20(i2,','))
 1002 format(16x,' *** DIVERGING SOLUTION ***',22x,20(i2,','))
      end
```

```
      SUBROUTINE SETUP
C
C *** Subroutine to set up matrices to be solved by gauss
C
      include 'bstlc:cmn'
C
C *** Rezero arrays
C
      do 40 i=1,n
      do 40 j=1,np1
   40 a(i,j)=0.
C
C *** Master DO loop to sum array values over all geophone 'hits'
C
      do 50 k=1,icount
C
C *** Define variables for each phone 'hit'.
C
        XF=gx-x(k)
        YF=gy-y(k)
        ZF=gz-z(k)
        TF=gt+t(k)
        D=sqrt(xf**2+yf**2+zf**2)
        F(1)=1.
        F(2)=-xf/(v*d)
        F(3)=-yf/(v*d)
        F(4)=-zf/(v*d)
        F(5)=d/v**2
        F(6)=gt+t(k)-d/v
C
C *** Set up a 5x6 matrix to be solved by subroutine gauss--
C       the first for an exactly determined solution (5 equations
C       5 unknowns), the second for an overdetermined set (icount
C       equations 5 unknowns).  The difdev is of
C       importance only for the second case (will be 0 for the
C       first).
C
        if (icount.eq.5) then
            do 48 j=1,np1
   48       a(k,j)=f(j)
          else
              do 49 i=1,n
            do 49 j=1,np1
   49       a(i,j)=f(i)*f(j)+a(i,j)
        endif
C
   50 continue
      return
      end
```

```
      SUBROUTINE GAUSS
C
C *** Subroutine to solve a set of linear equations--based on
C       gaussian elimination with back-substitution.
C
C *** Taken from:
C
C       Dorn, Wm., McCracken, D., 1972, Numerical Methods with
C         FORTRAN IV Case Studies, J. Wiley & Sons, p. 174.
C
      include 'bstlc.cmn'
      NM1 = N - 1
      DO 600 K = 1,NM1
      KP1 = K + 1
      L = K
C
C FIND TERM IN COLUMN K, ON OR BELOW MAIN DIAGONAL, THAT IS LARGEST
C   IN ABSOLUTE VALUE.  AFTER THE SEARCH, L IS THE ROW NUMBER OF
C   THE LARGEST ELEMENT.
C
      DO 400 I = KP1,N
400   IF (ABS(A(I,K)) .GT. ABS(A(L,K))) L = I
C
C CHECK WHETHER L = K, WHICH MEANS THAT THE LARGEST ELEMENT IN
C    COLUMN K WAS ALREADY THE DIAGONAL TERM, MAKING ROW INTERCHANGE
C    UNNECESSARY
C
      IF (L .EQ. K) GO TO 500
C
C INTERCHANGE ROWS L AND K, FROM DIAGONAL RIGHT
C
      DO 410 J = K, NP1
      TEMP = A(K,J)
      A(K,J) = A(L,J)
410   A(L,J) = TEMP
C
C ELIMINATE ALL ELEMENTS IN COLUMN K BELOW MAIN DIAGONAL
C ELEMENTS 'ELIMINATED' ARE NOT ACTUALLY CHANGED
C
500   DO 600 I = KP1,N
      FACTOR = A(I,K)/A(K,K)
      DO 600 J = KP1,NP1
600   A(I,J) = A(I,J) - FACTOR * A(K,J)
C
C BACK SOLUTION
C
      S(N) = A(N,NP1)/A(N,N)
      I = NM1
710   IP1=I+1
      SUM = 0.
      DO 700 J = IP1,N
700   SUM = SUM + A(I,J) * S(J)
      S(I) = (A(I,NP1) - SUM) / A(I,I)
      I = I - 1
      IF (I .GE. 1) GO TO 710
      RETURN
      END
```

```
      SUBROUTINE DEVCAL
C
C *** Subroutine to find the mixture of phones that gives the
C       minimum standard deviation (difmin) on calculated fit
C       (dropping 1 phone at a time), then send back the
C       original data.
C
      include 'bstlc.cmn'
C
      difmin=difdev
      itemp=icount
      icount=icount-1
C
C *** Sorting routine that drops 1 phone sequentially (j), and
C       computes the 'fit'
C
      do 60 j =1,icount
      do 50 i = 1,icount
      ij=i
      if(j.le.i)ij=i+1
      idsave(i)=id(ij)
      t(i)=torig(id(ij))
      x(i)=xloc(id(ij))
      y(i)=yloc(id(ij))
      z(i)=zloc(id(ij))
   50 continue
C
C *** Solve
C
      if(iguess.eq.1)call guess1
      if(iguess.eq.2)call guess2
      call newt
      if(idiv.eq.1)difdev=5000.
C
C *** Save the minumum phone no. as itemp
C
      if(difmin.gt.difdev.and.idiv.eq.0.and.iflag.eq.1)then
        itemp=j
        difmin=difdev
      else
      endif
   60 continue
C
C *** Reassign original values
C
      icount=icount+1
      do 100 i=1,icount
      idsave(i)=id(i)
      t(i)=torig(id(i))
      x(i)=xloc(id(i))
      y(i)=yloc(id(i))
      z(i)=zloc(id(i))
  100 continue
      return
      end
```

REFERENCES

Blake, W., 1983, "Microseismic Monitoring for Underground Stability," in Proc. 1st Int'l. Conf. on Stability in Underground Mining, ed. Brawner, SME of AIME, Denver, Colorado, pp. 272-294.

Blake, W., 1982, "Microseismic Applications for Mining: A Practical Guide," U.S. Bureau of Mines Open File Report 52-83, 206 pp.

Blake, W., Leighton, F., and Duvall, W., 1974, "Microseismic Techniques for Monitoring the Behavior of Rock Structures," U.S. Bureau of Mines Bulletin 665, 65 pp.

Blank, L., 1980, Statistical Procedures for Engineering, Management, and Science, McGraw-Hill Book Co., New York, 649 pp.

Carlisle, S.P., 1983, "New Microseismic Monitoring System Enables Rapid Analysis of Rock Burst Precursors at Hecla's Lucky Friday Mine," in Proc. 24th U.S. Symp. Rock Mech., 20-23 June 1983, Texas A&M, ed. Matheson, Assoc. of Eng. Geol., pp. 645-652.

Dahlquist, G., and Bjorck, A., 1974, Numerical Methods, Prentice-Hall, Inc., 573 pp.

Dorn, W., and McCracken, D., 1972, Numerical Methods with FORTRAN IV Case Studies, J. Wiley & Sons, Inc., New York, 447 pp.

James, M., Smith, G., and Wolford, J., 1977, Applied Numerical Methods for Digital Computation, 2nd ed., Harper & Row Pubs., Inc., New York, 687 pp.

Langstaff, J.T., 1979, "Computer-Based Microseismic Monitoring of Rock Burst Prone Areas," in Computer Methods for the 80's in the Mineral Industry, ed. Weiss, SME of AIME, Denver, Colorado, pp. 724-734.

Redfern, F.R., and Munson, R.D., 1982, "Acoustic Emission Source Location -- A Mathematical Analysis," U.S. Bureau of Mines RI 8692, 27 pp.

Van Eeckhout, E., Jordan, J., Hollis, J., and Hartmann, M., 1986, "Rockburst Monitoring at the Sunshine Mine, Kellogg, Idaho," in Proc. 19th APCOM, 14-16 Apr. 1986, Penn State, ed. Ramani, SME of AIME, Denver, Colorado, pp. 657-667.